【Spring学习35】Spring事务(5)：编程式事务

作者：soonfly (/authorarticle.html?author=soonfly) 2017-04-20 ☆ 收录到我的专题 (/select.html?articleId=401735)

标签 事务 (http://www.bijishequ.com/info/search.html?searchText=事务) 事务管理 (http://www.bijishequ.com/info/search.html?searchText=事务管理) TransactionTemplate (http://www.bijishequ.com/info/search.html?searchText=TransactionTemplate) 编程 (http://www.bijishequ.com/info/search.html?searchText=编程) Spring (http://www.bijishequ.com/info/search.html?searchText=Spring)

Spring提供了对编程式事务和声明式事务的支持，编程式事务允许用户在代码中精确定义事务的边界，而声明式事务（基于AOP）有助于用户将操作与事务规则进行解耦。简单地说，编程式事务侵入到了业务代码里面，但是提供了更加详细的事务管理；而声明式事务由于基于AOP，所以既能起到事务管理的作用，又可以不影响业务代码的具体实现。
Spring提供两种方式的编程式事务管理，分别是：使用**TransactionTemplate**和直接使用**PlatformTransactionManager**。

1、使用TransactionTemplate

采用TransactionTemplate和采用其他Spring模板，如JdbcTempalte和HibernateTemplate是一样的方法。它使用回调方法，把应用程序从处理取得和释放资源中解脱出来。如同其他模板，TransactionTemplate是线程安全的。代码片段：

```
TransactionTemplate tt = new TransactionTemplate(); // 新建一个TransactionTemplate
Object result = tt.execute(
new TransactionCallback(){
public Object doTransaction(TransactionStatus status){
updateOperation();
return resultOfUpdateOperation();
}
}); // 执行execute方法进行事务管理
```

使用TransactionCallback()可以返回一个值。如果使用TransactionCallbackWithoutResult则没有返回值。

2、使用PlatformTransactionManager

示例如下：

```
//定义一个某个框架平台的TransactionManager，如JDBC、Hibernate
DataSourceTransactionManager dataSourceTransactionManager = new DataSourceTransactionManager();

// 设置数据源
dataSourceTransactionManager.setDataSource(this.getJdbcTemplate().getDataSource());

// 定义事务属性
DefaultTransactionDefinition transDef = new DefaultTransactionDefinition();

// 设置传播行为属性
transDef.setPropagationBehavior(DefaultTransactionDefinition.PROPAGATION_REQUIRED);

// 获得事务状态
TransactionStatus status = dataSourceTransactionManager.getTransaction(transDef);
try {
// 数据库操作
dataSourceTransactionManager.commit(status);// 提交
} catch (Exception e) {
dataSourceTransactionManager.rollback(status);// 回滚
}
```